



日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2000年 8月15日

出願番号

Application Number:

特願2000-246404

出願人

Applicant(s):

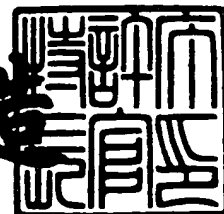
株式会社ソニー・コンピュータエンタテインメント

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 6月 5日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



【書類名】 特許願

【整理番号】 SCEI00019

【提出日】 平成12年 8月15日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 11/00
G06F 9/06

【発明の名称】 情報処理システム、実行可能モジュール生成方法および
記憶媒体

【請求項の数】 16

【発明者】
【住所又は居所】 東京都中野区中央一丁目3 8 番 1 号 住友中野坂上ビル
株式会社シュガーアンドロケッツ内

【氏名】 山本 浩

【発明者】
【住所又は居所】 東京都中野区中央一丁目3 8 番 1 号 住友中野坂上ビル
株式会社シュガーアンドロケッツ内

【氏名】 大平 俊充

【特許出願人】
【識別番号】 395015319

【氏名又は名称】 株式会社 ソニー・コンピュータエンタテインメント

【代理人】
【識別番号】 100084032

【弁理士】
【氏名又は名称】 三品 岩男

【電話番号】 045(316)3711

【選任した代理人】
【識別番号】 100087170

【弁理士】
【氏名又は名称】 富田 和子

【電話番号】 045(316)3711

【手数料の表示】

【予納台帳番号】 011992

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9912211

【ブルーフの要否】 要

【書類名】明細書

【発明の名称】情報処理システム、実行可能モジュール生成方法および記憶媒体

【特許請求の範囲】

【請求項 1】

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合によって作成される実行可能モジュールに前記プロテクトコードを含ませるコード書込み手段と、

前記復号後のプロテクト対象オブジェクトを、前記他のオブジェクトとの結合を行った後に消去する削除手段と、

を有することを特徴とする情報処理装置。

【請求項 2】

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成手段と、

前記復号後のプロテクト対象オブジェクトに前記第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールが作成されたら、当該実行可能モジュールに、前記第二のプロテクトコードを埋め込むコード書込み手段と、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトを、前

記二のプロテクトコード埋込み前に消去する削除手段とを有することを特徴とする情報処理装置。

【請求項 3】

請求項 2 記載の情報処理装置であって、

前記コード生成手段は、

乱数から、前記第一のプロテクトコードと前記第二のプロテクトコードを生成することを特徴とする情報処理装置。

【請求項 4】

請求項 2 または 3 記載の情報処理装置であって、

前記コード書込み手段は、

前記第一のプロテクトコードと前記第二のプロテクトコードとにダミーデータを付加することを特徴とする情報処理装置。

【請求項 5】

請求項 1、2、3 および 4 のいずれか 1 項に記載の情報処理装置であって、

前記コード書込み手段は、前記実行可能モジュールに含ませるためのプロテクトコードを暗号化し、

前記プロテクト対象オブジェクトは、前記実行可能モジュールに含まれている暗号化済みのプロテクトコードを、当該プロテクトコードの診断に際して復号する手続きを含む、

ことを特徴とする情報処理装置。

【請求項 6】

情報処理装置に処理を実行させるためのプログラムが格納された記憶媒体であって、

前記プログラムは、前記情報処理装置に、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

前記復号処理で生成されたプロテクト対象オブジェクトと他のオブジェクトとの結合により、前記実行可能モジュールを作成する結合処理と、

前記結合処理による作成される実行可能モジュールに前記プロテクトコードを含ませるためのコード書込み処理と、

前記復号処理で生成されたプロテクト対象オブジェクトを、前記他のオブジェクトとの結合後に消去する削除処理と、

を実行させることを特徴とする記憶媒体。

【請求項 7】

情報処理装置に処理を実行させるためのプログラムが格納された記憶媒体であって、

前記プログラムは、前記情報処理装置に、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成処理と、

前記復号処理後、当該復号処理で生成されたプロテクト対象オブジェクトに、前記第一のプロテクトコードを埋め込む第一コード書込み処理と、

前記コード書込み処理後、当該コード書込み処理で前記第一のプロテクトコードが埋め込まれたプロテクト対象オブジェクトと、他のオブジェクトとの結合により、実行可能モジュールを作成する結合処理と、

前記結合処理後、当該結合処理で作成された前記実行可能モジュールに前記第二プロテクトコードを埋め込む第二コード書込み処理と、

前記第一コード書込み処理後、前記第二コード書込み処理前に、前記復号処理で生成された前記プロテクト対象オブジェクトを消去する削除処理と、

を実行させることを特徴とする記憶媒体。

【請求項 8】

請求項 7 記載の記憶媒体であって、

前記プログラムは、前記コード生成処理において、前記情報処理装置に、

乱数から、前記第一のプロテクトコードと前記第二のプロテクトコードを生成させることを特徴とする記憶媒体。

【請求項 9】

請求項 7 または 8 記載の記憶媒体であって、
前記プログラムは、前記情報処理装置に、
前記第一のプロテクトコードと前記第二のプロテクトコードとにダミーデータを付加させることを特徴とする記憶媒体。

【請求項 10】

請求項 6、7、8 および 9 のいずれか 1 項に記載の記憶媒体であって、
前記プログラムは、前記情報処理装置に、
前記実行可能モジュールに含ませるためのプロテクトコードを暗号化する処理を実行させ、

前記プロテクト対象オブジェクトには、前記実行可能モジュールに含まれている暗号化プロテクトコードを、当該プロテクトコードの診断に際して復号する手続きが含まれ、

ることを特徴とする記憶媒体。

【請求項 11】

情報処理装置によって処理されるオブジェクトを記憶した記憶媒体であって、
暗号化されたプロテクト対象オブジェクトが格納され、
当該プロテクト対象オブジェクトは、
自身を組み込んだ実行可能モジュールに含まれた 1 以上のプロテクトコードに矛盾があれば処理を終了させる手続きを含む

ことを特徴とする記憶媒体。

【請求項 12】

請求項 11 記載の記憶媒体であって、
前記プロテクト対象オブジェクトは、
前記実行可能モジュールに含まれているプロテクトコードが暗号化されている場合に、当該プロテクトコードの診断に先立ち、当該プロテクトコードを復号する手続きを含む、

ことを特徴とする記憶媒体。

【請求項 13】

複数のオブジェクトを結合して、実行可能モジュールを生成する処理を情報処理装置に実行させる実行可能モジュール生成方法であって、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合と、前記プロテクトコードの書込みとによって、前記実行可能モジュールを生成し、

前記復号後のプロテクト対象オブジェクトは、前記他のオブジェクトとの結合がなされた後に消去される、

ことを特徴とする実行モジュール生成方法。

【請求項 1 4】

複数のオブジェクトを結合して、実行可能モジュールを生成する処理を情報処理装置に実行させる実行可能モジュール生成方法であって、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成して、前記復号後のプロテクト対象オブジェクトに当該第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールを作成したら、当該実行可能モジュールに当該第二のプロテクトコードを埋め込み、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトは、前記二のプロテクトコード埋込み前に消去されることを特徴とする実行可能モジュール生成方法。

【請求項 1 5】

複数のオブジェクトの結合により組み立てられた実行可能モジュールを実行可能な装置が実行するための実行可能モジュールが格納された記憶媒体であって、

前記複数のオブジェクトには、

少なくとも 1 つのプロテクトコードに矛盾があるか否かを診断し、当該診断結

果に応じて、前記実行可能モジュールの処理を終了させる手続きを含んだライブラリオブジェクトが含まれ、

前記実行可能モジュールには、

少なくとも1つのプロテクトコードが埋め込まれている、

ことを特徴とする記憶媒体。

【請求項 1 6】

複数のオブジェクトの結合によって生成された実行可能モジュールを実行するエンタテインメント装置であって、

前記少なくとも1つのオブジェクトに含まれたプロテクトコードと、それ以外のプロジェクトコードとが前記実行可能モジュールに含まれている場合、それらのプロテクトコード間の関係を診断し、当該関係に矛盾があれば、前記実行可能モジュールの処理を終了させる手段を有することを特徴とするエンタテインメント装置。

【発明の詳細な説明】

【 0 0 0 1】

【発明の属する技術分野】

本発明は、プログラム開発環境等におけるリソース不正コピーの防止技術に関する。

【 0 0 0 2】

【従来の技術】

プログラムの不正コピーを防止するための技術として、特許公報第 2 5 6 9 5 6 4 号記載の、ソフトウェアのコピープロテクト装置が知られている。この装置によれば、フロッピディスクから読み出したプロテクトコードを外部出力しないため、その不正な書替えを防止することができる。これにより、不正に複製されたソフトウェアを実行可能とすることが困難となり、ソフトウェアの不正な複製を効果的に防止することができる。

【 0 0 0 3】

【発明が解決しようとする課題】

プログラムが作成される場合には、通常、グラフィック機能その他の基本的機

能を提供するライブラリ等のリソースが使用される。このようなライブラリ等のリソースに含まれているオブジェクトは、実行可能モジュールではない。そのため、不正コピー防止のためのプロテクトをライブラリにかけることは困難である。したがって、現在のところ、ライブラリ等のリソースは、メーカ側がプログラム開発者の良識を信頼するというかたちで、プロテクトがかけられていないまま提供されているのが実情である。

【 0 0 0 4 】

しかし、ライブラリ等のリソースは、プログラムと同様、重要な開発成果であるため、なんらかのコピープロテクトがかけられるようにすることが望まれている。

【 0 0 0 5 】

そこで、本発明は、実行可能モジュール以外のオブジェクトの不正コピーを防止することを目的とする。

【 0 0 0 6 】

【課題を解決するための手段】

上記課題を解決するため、本発明は、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合によって作成される実行可能モジュールに前記プロテクトコードを含ませるコード書込み手段と、

前記復号後のプロテクト対象オブジェクトを、前記他のオブジェクトとの結合を行った後に消去する削除手段と、

を有することを特徴とする情報処理装置を提供する。

【 0 0 0 7 】

また、本発明は、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成手段と、

前記復号後のプロテクト対象オブジェクトに前記第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールが作成されたら、当該実行可能モジュールに、前記第二のプロテクトコードを埋め込むコード書込み手段と、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトを、前記第二のプロテクトコード埋込み前に消去する削除手段とを有することを特徴とする情報処理装置を提供する。

【 0 0 0 8 】

また、これらの情報処理装置によって作成された実行可能モジュールは、記憶媒体に格納することができる。例えば、

複数のオブジェクトの結合により組み立てられた実行可能モジュールを実行可能な装置が実行するための実行可能モジュールが格納された記憶媒体であって、

前記複数のオブジェクトには、

少なくとも1のプロテクトコードに矛盾があるか否かを診断し、当該診断結果に応じて、前記実行可能モジュールの処理を終了させる手続きを含んだライブラリオブジェクトが含まれ、

前記実行可能モジュールには、

少なくとも1のプロテクトコードが埋め込まれている、

ことを特徴とする記憶媒体とすることができる。

【 0 0 0 9 】

ここで、プロテクトコードは、乱数から生成されるようにしてよい。また、プ

ロテクトコードにダミーデータが付加されるようにしてもよい。また、実行可能モジュールに含まれているプロテクトコードを復号する手続きをプロテクト対象オブジェクトに含ませておけば、プロテクトコードを暗号化してから実行可能モジュールに含ませるようにすることもできる。

【 0 0 1 0 】

さらに、本発明は、これらの情報処理装置に処理を実行させるため情報が格納された記憶媒体として、

(1) 情報処理装置に処理を実行させるためのプログラムが格納された記憶媒体であって、

前記プログラムは、前記情報処理装置に、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

前記復号処理で生成されたプロテクト対象オブジェクトと他のオブジェクトとの結合により、前記実行可能モジュールを作成する結合処理と、

前記結合処理による作成される実行可能モジュールに前記プロテクトコードを含ませるためのコード書込み処理と、

前記復号処理で生成されたプロテクト対象オブジェクトを、前記他のオブジェクトとの結合後に消去する削除処理と、

を実行させることを特徴とする記憶媒体、

(2) 情報処理装置に処理を実行させるためのプログラムが格納された記憶媒体であって、

前記プログラムは、前記情報処理装置に、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成処理と、

前記復号処理後、当該復号処理で生成されたプロテクト対象オブジェクトに、

前記第一のプロテクトコードを埋め込む第一コード書込み処理と、

前記コード書込み処理後、当該コード書込み処理で前記第一のプロテクトコードが埋め込まれたプロテクト対象オブジェクトと、他のオブジェクトとの結合により、実行可能モジュールを作成する結合処理と、

前記結合処理後、当該結合処理で作成された前記実行可能モジュールに前記第二プロテクトコードを埋め込む第二コード書込み処理と、

前記第一コード書込み処理後、前記第二コード書込み処理前に、前記復号処理で生成された前記プロテクト対象オブジェクトを消去する削除処理と、

を実行させることを特徴とする記憶媒体、

(3) 情報処理装置によって処理されるオブジェクトを記憶した記憶媒体であって

暗号化されたプロテクト対象オブジェクトが格納され、

当該プロテクト対象オブジェクトは、

自身を組み込んだ実行可能モジュールに含まれた 1 以上のプロテクトコードに矛盾があれば処理を終了させる手続きを含む

ことを特徴とする記憶媒体、

を提供する。

【0011】

ここで、プロテクト対象オブジェクトは、ライブラリ、画像データ等、実行可能モジュール以外のオブジェクトであってもよい(以下、同じ)。

【0012】

また、本発明は、

複数のオブジェクトを結合して、実行可能モジュールを生成する処理を情報処理装置に実行させる実行可能モジュール生成方法として、

(1) 実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合と、前記プロテクトコードの書込みとによって、前記実行可能モジュールを生成し、

前記復号後のプロテクト対象オブジェクトは、前記他のオブジェクトとの結合がなされた後に消去される、

ことを特徴とする実行モジュール生成方法、および、

(2) 実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成して、前記復号後のプロテクト対象オブジェクトに当該第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールを作成したら、当該実行可能モジュールに当該第二のプロテクトコードを埋め込み、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトは、前記二のプロテクトコード埋込み前に消去されることを特徴とする実行可能モジュール生成方法を提供する。

【 0 0 1 3 】

この実行可能モジュール生成方法により生成された実行可能モジュールを実行する装置の 1 つとして、本発明は、

複数のオブジェクトの結合によって生成された実行可能モジュールを実行するエンタテインメント装置であって、

前記少なくとも 1 つのオブジェクトに含まれたプロテクトコードと、それ以外のプロジェクトコードとが前記実行可能モジュールに含まれている場合、それらのプロテクトコード間の関係を診断し、当該関係に矛盾があれば、前記実行可能モジュールの処理を終了させる手段を有することを特徴とするエンタテインメント装置を提供する。

【 0 0 1 4 】

なお、ここで使用されている各符合(1)(2)(3)は、パラグラフの整理のために付したものの過ぎない。

【 0 0 1 5 】

【発明の実施の形態】

以下、添付の図面を参照しながら、本発明に係る実施の一形態について説明する。ただし、ここでは、ライブラリをプロテクト対象とする。

【 0 0 1 6 】

まず、プログラム開発側で使用される情報処理システムの概略ハードウェア構成について説明する。

【 0 0 1 7 】

プログラム開発側の情報処理システムは、図 1 に示すように、情報処理装置 1 0 0、ユーザ(プログラム開発者)からのデータ入力を受け付ける入力装置(マウス、キーボード等) 1 2 0、各種データを出力する出力装置(ディスプレイ装置等) 1 3 0、等から構成されている。ただし、このシステムには、その他の装置が必要に応じて増設されることもある。

【 0 0 1 8 】

情報処理装置 1 0 0 は、後述の実行可能プログラムファイル作成処理等を実行する CPU 1 0 1、グラフィック描画処理等を実行するグラフィックス機構 1 0 2、主記憶 1 0 3、CPU 1 0 1 と主記憶 1 0 3 との間をつないだプロセッサバス 1 0 5、主記憶 1 0 3 またはバス 1 0 6 と CPU 1 0 1 等との間のデータ転送を制御するメモリ/バス制御チップ 1 0 4、グラフィックス機構 1 0 2 のグラフィックス制御チップとメモリ/バス制御チップ 1 0 4 との間をつないだ AGP、ソフトウェア開発のために必要となる各種データ(シリアル番号等の識別情報が埋め込まれたフロントエンドツール、フロントエンドツールによって起動されるリンカおよびコンパイラ、フロントエンドツールとともに提供された暗号化ライブラリファイル等)が格納されたハードディスク 1 0 6、ハードディスク 1 0 6 に対するデータ読み込み/書き込み処理を実行するハードディスク・ドライブ 1 0 7、ハードディスクドライブ 1 0 7 がつながれた高速バス 1 0 6、記憶媒体 A に対するデータ読み込み読書き/書き込み処理を実行するドライブ 1 1 2、入力装置 1 2 0 からのデータ転送を制御するコントローラ 1 1 1、出力装置 1 3 0 へのデータ転送を制御するコントローラ 1 1 3、各コントローラ 1 1 1, 1 1 3 とドライブ 1 1 2 とがつながれた低速バス 1 0 9、低速バス 1 0 9 と高速バス 1 0 6 との間をつないだブリッジ回路 1 1 0、通信ポート(シリアルポート、パラレルポート

等)、等を備えている。なお、ここで示した情報処理装置100の内部構成は、ソフトウェア開発に使用されるものとしての一例である。

【0019】

つぎに、この情報処理システムのハードディスク106内の格納データについて説明する。

【0020】

コンパイラは、CPUに、指定ソースプログラムファイルに記述されたソースの解析処理(字句解析、構文解析、意味解析)等によって、オブジェクトとして翻訳編集させ、それをオブジェクトファイルに格納させるための処理が定義されたソフトウェアである。リンカは、CPUに、指定オブジェクトファイル群のそれぞれに記述されているオブジェクトを結合編集させることによって実行可能モジュールを組み立させ、それを実行可能プログラムファイルに格納させるための処理が定義されたソフトウェアである。そして、本実施の形態に係るフロントエンドツールは、CPUに、コンパイラおよびリンカを起動させ、後述の実行可能プログラムファイル作成処理を実行させるための処理が定義されたソフトウェアである。

【0021】

暗号化Libファイル L_1 は、フロントエンドツールとともに、ユーザ(プログラム開発者)に提供される。例えば、暗号化Libファイル L_1 が記録されたインストール用CD-ROMと、フロントエンドツールが記録されたインストール用フロッピーディスクとがワンセットとしてユーザに提供される。

【0022】

この暗号化Libファイル L_1 は、適当な暗号化処理によって1以上のLibファイルを暗号化することによって作成されたものである。ユーザは、実行可能プログラムファイル作成時に、この暗号化Libファイル L_1 をフロントエンドツールで復号化することによって、実行可能プログラムファイル作成に必要な1以上のLibファイル L_2 が得ることができる。このように、ライブラリを暗号化された状態で提供し、それと共に提供されたフロントエンドツールによってその復号化を行なうようにしたのは、フロントエンドツールを持たないユーザによ

る L i b ファイル不正使用を禁じるためである(後述)。

【 0 0 2 3 】

そして、暗号化 L i b ファイル L_1 の復号により生成される L i b ファイル L_2 には、グラフィック機能、イベント処理等、アプリケーションにおいて使用される基本的な関数等のオブジェクトが部品化されて格納されている。また、すべてのアプリケーションにおいて呼び出される初期化関数が含まれている L i b ファイル内オブジェクトには、さらに、初期化関数において呼び出されるプロテクトコードチェック手続きと、オブジェクトファイル M_2 との結合編集前に第一プロテクトコード C_1 が格納される第一プロテクトコード格納変数とが定義されている。

【 0 0 2 4 】

ここで、プロテクトコードチェック手続きには、第一プロテクトコード格納変数の格納データとプログラムファイル末尾の格納データとが、2つのプロテクトコード C_1, C_2 の満たすべき関係を満たしているか否かをチェックし、その関係を満たさなければ実行中プログラムを初期化段階で終了させる手続きが定義されている。

【 0 0 2 5 】

なお、本実施の形態では、初期化関数が含まれている L i b ファイル L_2 だけにプロテクトコードチェック関数および第一プロテクトコード格納変数を含ませているが、すべての L i b ファイルにプロテクトコードチェック関数および第一プロテクト格納変数を含ませることにしてもよいし、代表的な L i b ファイル、例えば、プロテクトをかける必要のある L i b ファイルだけにプロテクトコードチェック関数および第一プロテクト格納変数を含ませることにしてもよい。

【 0 0 2 6 】

本実施の形態に係る情報処理システムは、このようなハードウェア構成およびハードディスク格納データ(フロントエンドツール、コンパイラ、リンカ等のソフトウェア、暗号化ライブラリファイル)によって、図2に示すような機能構成を実現する。

【 0 0 2 7 】

具体的には、(1)フロントエンドツールの起動コマンドの入力を受け付ける入力受付部 2 0 0、(2)起動時にフロントエンドツール複製チェック処理を実行する起動時チェック処理部 2 0 1、(3)開発成果(ソースプログラムファイル M_1 、オブジェクトファイル M_2 、実行可能プログラムファイル M_3)を格納するための開発成果記憶部 2 0 2、(4)ソースプログラムファイル M_1 からオブジェクトファイル M_2 を生成するコンパイル処理部 2 0 3、(5)暗号化ライブラリファイル L_1 (暗号化 $L i b$ ファイル L_1 と呼ぶ)を記憶したライブラリ記憶部 2 0 4、(6)暗号化 $L i b$ ファイル L_1 を複号する復号化処理部 2 0 5、(7)復号化処理部 2 0 4 の復号化処理によって生成されたライブラリファイル($L i b$ ファイルと呼ぶ)を一時ファイル L_2 として記憶する一時ファイル記憶部($t m p$) 2 0 6、(8) $L i b$ ファイル L_2 とオブジェクトファイル M_2 との結合編集により実行可能プログラムファイル M_3 を生成するリンク処理部 2 0 7、(9)リンク処理部 2 0 7 の結合編集処理後に一時ファイル L_2 を削除する一時ファイル削除処理部 2 0 8、(10)リンク処理部 2 0 7 の結合編集処理前の $L i b$ ファイル L_2 とこの $L i b$ ファイル L_2 から作成された実行可能プログラムファイル M_3 とにそれぞれプロテクトコード C_1, C_2 を付与するコード付与処理部 2 0 9、(11)これら各処理部を制御する制御処理部 2 1 0、が実現される。

【 0 0 2 8 】

なお、コード付与処理部 2 0 9 は、(12)互いに関連付けられた 1 組のプロテクトコード C_1, C_2 (例えば、 $C_1 = f(C_2)$ の関係を満たす 1 組のプロテクトコード C_1, C_2)を生成するコード生成処理部 2 0 9 a と、(13)コード生成処理部 2 0 9 a が生成した 1 組のプロテクトコード C_1, C_2 のうち一方のコード C_1 (第一プロテクトコードと呼ぶ)を $L i b$ ファイル L_2 内のコード格納変数に書き込むとともに他方のコード C_2 (第二プロテクトコードと呼ぶ)を実行可能プログラムファイル M_3 の末尾に付加するコード書込処理部 2 0 9 b とからなっている。

【 0 0 2 9 】

つぎに、これら各機能構成部により実行される実行可能プログラムファイル作成処理について説明する。

【 0 0 3 0 】

プログラム開発者は、コーディングが終了したら、ソースプログラムファイル M_1 をハードディスク 1 0 8 の所定の領域(図 2 の開発成果記憶部 2 0 2 に相当)に格納してから、そのソースプログラムファイル名と必要なライブラリファイル名とを指定してフロントエンドツールを起動する。これにより、CPU が、フロントエンドツールに定義された処理を実行し、以下の実行可能プログラムファイル作成処理(図 2 のフローチャート参照)が実現する。

【 0 0 3 1 】

まず、フロントエンドツールの起動コマンドを受け付けた入力受付部 2 0 0 は、制御処理部 2 1 0 に処理開始を指示する(S 3 0 0)。この指示を受け付けると、制御処理部 2 1 0 は、フロントエンドツール複製チェック処理の実行を起動時チェック処理部 2 0 1 に指示する。

【 0 0 3 2 】

この指示にしたがって、起動時チェック処理部 2 0 1 は、以下に示す方法により、フロントエンドツールの不正使用をチェックする(S 3 0 1)。

【 0 0 3 3 】

例えば、プログラム開発環境に LAN が構築されており、ネットワーク上で起動中のプログラムの識別情報が重複していないか否かを監視するチェックプログラムがネットワーク上に常駐している場合であれば、起動時チェック処理部 2 0 1 は、フロントエンドツールに付与されている識別情報をネットワーク上に送信する。これに対して、チェックプログラムは、それと同じ識別情報がネットワーク上に存在するか否かを返信する。なお、フロントエンドツールがコピー品であれば、同じ識別情報が複数ネットワーク上に存在しており、フロントエンドツールがコピー品でなければ、同じ識別情報が複数ネットワーク上に存在することはない。

【 0 0 3 4 】

また、フロントエンドツールに dongle が割り当てられている場合であれば、起動時チェック処理部 2 0 1 は、情報処理装置の通信ポートに装着されている dongle に認証メッセージを送る。これに対して、dongle は、自身に割り当てられた識別情報を返信する。なお、認証されたユーザがフロントエンドツールを起

動した場合であれば、 dongle から返信された識別情報が、フロントエンドツールの識別情報と一致し、認証されたユーザ以外のユーザがフロントエンドツールを起動した場合であれば、 dongle の装着がないために識別情報の返信がないか、または、 dongle から返信された識別情報がフロントエンドツールの識別情報と一致しないことになる。

【 0 0 3 5 】

このようなチェックの結果に基づき、起動時チェック処理部 2 0 1 は、フロントエンドツールの使用に不正(フロントエンドツールが複製品である、認証ユーザ以外のユーザによる使用である、等)があるか否かを判断する(S 3 0 2)。

【 0 0 3 6 】

その結果、起動時チェック処理部 2 0 1 がフロントエンドツールの使用に不正があると判断した場合、制御処理部 2 1 0 が、エラーメッセージを出力してから(S 3 0 3)、処理を終了させる(S 3 0 8)。これにより、フロントエンドツールの不正使用が禁止されるため、フロントエンドツールの不正使用者には、暗号化 Lib ファイルを復号することが困難となる。したがって、Lib ファイルの不正使用も禁止される。

【 0 0 3 7 】

一方、起動時チェック処理部 2 0 1 がフロントエンドツールの使用に不正がないと判断した場合、制御処理部 2 1 0 が、指定のソースプログラムファイル M₁ の機械語翻訳をコンパイル処理部 2 0 3 に指示する。

【 0 0 3 8 】

この指示に応じて、コンパイル処理部 2 0 3 は、指定のソースプログラムファイル M₁ をアセンブリプログラムにコンパイルし、そのアセンブリプログラムをオブジェクトにアセンブルする。そして、ここで生成されたオブジェクトファイル M₂ を開発成果記憶部 2 0 2 に格納してから、制御処理部 2 1 0 に正常終了を報告する。

【 0 0 3 9 】

制御処理部 2 1 0 は、コンパイル処理部 2 0 3 から正常終了報告を受け付けると、Lib ファイルの復号化を復号化処理部 2 0 5 に指示する。

【 0 0 4 0 】

この指示に応じて、復号化処理部 2 0 5 は、ライブラリ記憶部 2 0 4 から暗号化 L i b ファイル L_1 を取り出し、それを復号化する。そして、ここで生成された 1 以上の L i b ファイル L_2 を、適当なファイル名で、一時ファイル記憶部 2 0 6 に一時ファイルとして格納してから、制御処理部 2 1 0 に正常終了を報告する。

【 0 0 4 1 】

制御処理部 2 1 0 は、復号化処理部 2 0 5 からの正常終了報告を受け付けると、L i b ファイル L_2 に対するプロテクトコード付与をコード付与処理部 2 0 9 に指示する。

【 0 0 4 2 】

この指示に応じて、コード付与処理部 2 0 9 では、まず、コード生成処理部 2 0 9 a が、乱数 C を発生する。そして、図 4 に示すように、この乱数 C を 2 つの加工方法 F_1, F_2 (例えば、乱数 C を入力データとする 2 つの関数等) でそれぞれ加工することによって 2 つのプロテクトコード C_1, C_2 を生成し、さらに、これらのプロテクトコード C_1, C_2 をそれぞれ適当な暗号化方法によって暗号化する。ついで、コード書込み処理部 2 0 9 b が、第二プロテクトコード C_2 を保持するとともに、所定の L i b ファイル L_2 内の第一プロテクトコード格納変数に第一プロテクトコード C_1 を格納する (S 3 0 4)。なお、ここで L i b ファイル L_2 に埋め込まれた第一プロテクトコード C_1 の暗号化前のコードと、コード書込み処理部 2 0 9 b によって保持されている第二プロテクトコード C_2 の暗号化前のコードとの間には、前述したように、予め定められた関係(ここでは、 $C_1 = f(C_2)$ とする)がある。

【 0 0 4 3 】

L i b ファイル L_2 に対するプロテクトコード付与が終了すると、制御処理部 2 1 0 は、リンク処理部 2 0 7 に、オブジェクトファイル M_2 と指定の L i b ファイル L_2 との結合編集を指示する。

【 0 0 4 4 】

この指示に応じて、リンク処理部 2 0 7 は、オブジェクトファイル M_2 と指定

のLibファイル L_2 とを結合し、実際の番地の埋込み等を行う。そして、これにより組み立てられた実行可能プログラムファイル M_3 を、適当な名前で開発成果記憶部202に格納してから、正常終了を制御処理部210に報告する(S305)。

【0045】

制御処理部210は、リンク処理部207からの正常終了報告を受け付けると、一時ファイル削除を一時ファイル削除処理部208に指示する。この指示に応じて、一時ファイル削除処理部208は、すべてのLibファイル L_2 を一時ファイル記憶部206から削除する(S306)。

【0046】

その後、制御処理部210は、実行可能プログラムファイル L_3 に対するプロテクトコード付与をコード付与処理部209に指示する。

【0047】

この指示に応じて、コード付与処理部209では、コード書込み処理部209bが、保持中の第二プロテクトコード C_2 を実行可能プログラムファイル M_3 に付加する。そして、正常終了を制御処理部210に報告する(S307)。なお、ライセンシー管理(例えば、プログラム作成会社の識別等)のために、コード付与処理部209が、このとき、さらに、フロントエンドツールの識別情報を実行可能プログラムファイル M_3 に埋め込まむようにしてもよい。

【0048】

制御処理部210は、コード付与処理部209からの正常終了報告を受け付けると、処理を終了させる(S308)。なお、以上の処理により作成された実行可能プログラムファイル M_3 は、例えば、光ディスク等の記憶媒体に焼き付けられて市場に供給される(図9参照)。

【0049】

本実施の形態に係る実行可能プログラムファイル作成処理によれば、以下に示すように、Libファイルの不正使用が防止される。

(1)ハードディスクに格納されたLibファイルが暗号化されており、かつ、それを復号するフロントエンドツールの不正使用が、起動時チェック(図2のS3

01)によって禁止される。このため、フロントエンドツールの正当使用者以外のユーザによるLibファイル使用が防止される。

(2)図5に示すように、実行中のプログラムファイルに埋め込まれたプロテクトコードに矛盾があった場合にその処理を終了させるプロテクトコードチェック手続きEをLibファイル L_2 にあらかじめ含めておくとともに、フロントエンドツールの使用によってLibファイル L_2 とオブジェクトファイル M_2 とが結合された実行可能プログラムファイル M_3 にだけ、実行可能プログラムファイル作成処理の過程(S304、S307)において適正なプロテクトコード C_1, C_2 が埋め込まれるようにすることによって、フロントエンドツールの正当使用により作成された実行可能プログラムファイル M_3 のみを実行可能とすることができる。

すなわち、フロントエンドツールによらずLibファイルが結合された実行可能プログラムファイル、具体的には、不正なプロテクトコード(一方または両方)が埋め込まれた実行可能プログラムファイル、プロテクトコード(一方または両方)が埋め込まれていない実行可能プログラムファイル等の実行が、初期化段階において阻止される。このため、フロントエンドツールの正当使用者以外のユーザによるLibファイル使用が防止される。

(3) 図5に示すように、乱数Cから生成された1組のプロテクトコード C_1, C_2 のうち、第一プロテクトコード C_1 だけをLibファイル L_2 に埋め込んでおき(S304)、その後、Libファイル L_2 とオブジェクトファイル M_2 との結合編集(S305)が終了したら、実行可能プログラムファイル M_2 への第二プロテクトコード C_2 の埋込み前にLibファイル L_2 を消去(S306)してしまうため、使用済みLibファイル L_2 の不正利用が防止される。すなわち、一方のプロテクトコード C_1 が埋め込まれた使用済みLibファイル L_2 が消去前に取り出されたとしても、乱数Cから生成された他方のプロテクトコード C_2 の取り出しがその段階(S304～S306)では不可能であるから、その使用済みLibファイル L_2 に埋め込まれている第一プロテクトコード C_1 と矛盾しない第二プロテクトコード C_2 を知ることは困難である。このため、消去前に取り出した使用済みLibファイル L_2 とオブジェクトファイルとを結合編集したとしても、使用済みLibファイル L_2 に埋込済みの第一プロテクトコードと適正な関係を有する第

ニプロテクトコードを実行可能プログラムファイルに埋め込むことは困難である。したがって、使用済みLibファイルの不正使用が防止される。

(4)最終的に作成された実行可能プログラムファイル M_3 の解析によって、それに埋め込まれた1組のプロテクトコード C_1, C_2 を知ってから、つぎのフロントエンドツール起動時に使用済みLibファイル L_2 を消去前に取り出したとしても、プロテクトコード C_1, C_2 は、毎回、乱数から生成されるため、消去前に取り出した使用済みLibファイル L_2 に埋込済みの第一プロテクトコード C_1 に対応する第二プロテクトコードは、実行可能プログラムファイル M_3 の解析によって知った第二プロテクトコード C_2 とは必ずしも一致していない。このため、最終的に作成された実行可能プログラムファイル M_3 の解析によっても、適正な関係を有する1組のプロテクトコードを得ることは困難である。したがって、消去前に取り出した使用済みLibファイル L_2 とオブジェクトファイルとを結合編集したとしても、使用済みLibファイル L_2 に埋込済みの第一プロテクトコードと適正な関係を有する第二プロテクトコードを実行可能プログラムファイルに埋め込むことは困難である。このため、使用済みLibファイルの不正使用が防止される。

【0050】

以上の4つの防止策によってLibファイル不正使用が防止されるが、ライブラリバージョンアップのたびに、Libファイルの暗号化方法とプロテクトコードチェック手続きの内容とをそれぞれ変更するようにすれば、Libファイルの不正使用防止が強化される。なお、このようにする場合には、Libファイル複合方法および乱数加工方法が変更されたフロントエンドツールが、新バージョンの暗号化Libファイルとともにリリースされる必要がある。

【0051】

ところで、以上においては、Libファイル内 L_2 で定義された第一プロテクトコード格納変数に第一プロテクトコード C_1 を格納し、実行可能プログラムファイル M_3 の末尾に第二プロテクトコード C_2 を付加することとしているが、必ずしも、このようにする必要はない。例えば、プロテクトコード付与処理部が、図6に示すように、Libファイル L_2 の末尾に、ダミーデータと第一プロテクト

コード C_1 とを含んだ第一ダミーエリア①を付加し、実行可能プログラムファイル M_3 の末尾に、ダミーデータと第二プロテクトコード C_2 とを含んだ第二ダミーエリア②を付加するようにしてもよい。そして、2つのダミーエリア①②の合計データ量を変えないように、各ダミーエリア①②に含まれるダミーデータのデータ量をランダムに変化させるようにすれば、1つの実行可能プログラムファイル M_3 を作成するごとに、その内部におけるプロテクトコード C_1, C_2 のアドレスが変化するため、プロテクトコード C_1, C_2 の間にある関係の解析を困難にすることができる。

【 0 0 5 2 】

また、以上においては、プロテクトコードチェック手続きが組み込まれたLibファイルに第一プロテクトコードをリンク前に埋め込み、実行可能プログラムファイルが完成したときに、さらに、この実行可能プログラムファイルに第二プロテクトコードを埋め込むこととしているが、かならずしも、このようにする必要はない。例えば、プロテクトコードチェック手続きが組み込まれたLibファイルにはプロテクトコードを組み込まず、実行可能プログラムファイルが完成したときに、この実行可能プログラムファイルに、プロテクトコードチェック手続きによって正当性をチェック可能なプロテクトコード(例えば、互いに関連付けられた複数のプロテクトコード)を埋め込むようにしてもよい。

【 0 0 5 3 】

つぎに、本実施の形態に係るプログラムファイル作成処理により作成された実行可能プログラムファイル M_3 を実行することができるシステムの概略ハードウェア構成について説明する。ただし、ここでは、システムの一例として、エンタテインメント装置を挙げることにする。

【 0 0 5 4 】

図7は、本実施の形態に係るエンタテインメント装置700およびその周辺機器の外観図であり、図8は、本実施の形態に係るエンタテインメント装置700のハードウェア構成図である。

【 0 0 5 5 】

本実施の形態に係るエンタテインメント装置700は、CD-ROM、DVD

ーROM等の光ディスクからゲームプログラム(前述のプログラムファイル作成処理により作成された実行可能プログラムファイルM₃等)を読み出し、そのゲームプログラムにしたがってゲームを実行するものである。ここで、「ゲームの実行」とは、エンタテインメント装置700が、ユーザ(ゲームプレイヤ)からの指示にしたがって、表示装置(CRT、LCD、プロジェクション装置等)上の表示映像およびオーディオ装置からの音声をゲームストーリー等にあわせて変化させることをいう。

【0056】

図7に示すように、エンタテインメント装置700の筐体前面部には、ユーザ(ゲームプレイヤ)がゲームを実行させるために必要となる各種の操作部が設けられている。このような操作部の具体例としては、ゲームプログラムが格納された光ディスクが装着されるディスク装着部701、ディスク装着部701からトレイを引き出すためのトレイ操作ボタン701A、ユーザ(ゲームプレイヤ)からの入力を受け付ける各種入力受け付け部(ボタン753A, 753B, 754~758、方向キー752、スティック751A, 751B等)を有するコントローラ750が接続される複数(ここでは2つ)のコントローラ接続部702A, 702B、実行中のゲームをリセットするためのリセットボタン703、ゲームデータ等を記録させるためのメモ리카ード730が装着される複数(ここでは2つ)のメモ리카ード装着部704A, 704B、等が挙げられる。なお、ここで、コントローラ接続部702A, 702Bを複数設けているのは、コントローラ接続部702A, 702Bにそれぞれコントローラを接続することによって、エンタテインメント装置700が、複数のゲームプレイヤからの指示を受け付けることができるようにするためである。

【0057】

また、エンタテインメント装置700の筐体背面部(前面の反対側)には、電源オンオフスイッチ(不図示)、表示装置等が接続されるAV端子(不図示)が設けられている。

【0058】

そして、このエンタテインメント装置700の内部には、図8に示すように、

装置全体を制御するメインCPU 800、AV端子(不図示)から出力されるビデオ信号を生成するグラフィックプロセッサ(GP) 801、コントローラ接続部702A, 702Bに接続されたコントローラ750およびメモ리카ード接続部704A, 704Bに接続されたメモ리카ード730のデータ入出力を制御するI/Oプロセッサ(IOP) 802、ディスク装着部701に装着された光ディスクからのデータ読出しを制御する光ディスク制御部803、AV端子(不図示)から出力されるオーディオ信号を生成するサウンド再生処理プロセッサ(SPU) 804、サウンドバッファ805、メインCPU 800およびIOP 802が実行する各オペレーティングシステムプログラムが格納されたOS-ROM 806、IOPのワークエリアとして用いられるIOPメモリ807、メインCPU 800のワークエリアおよびバッファとして用いられるメインメモリ808、これら各部の間をつなぐバス809等が搭載されている。

【0059】

このエンタテインメント装置700に電源が投入されると、2種類のオペレーティングシステムプログラム(メインCPU用、IOP用)がそれぞれOS-ROM 806から読みだされ、それらがメインCPU 800とIOP 802とによってそれぞれ起動される。これにより、オペレーティングシステムによって、装置各部の統合制御が開始され、エンタテインメント装置としての各種機能がユーザ(ゲームプレイヤー)に提供される。具体的には、光ディスクからの実行可能プログラムファイル読込みおよびその実行、コントローラを介したプレイヤーからの指示受付、プレイヤーの指示に応じた映像表示および効果音・楽音出力等が可能な環境が提供される。

【0060】

このような環境において、図9に示した光ディスクから、上述の実行可能プログラムファイル作成処理によって作成されたゲーム用実行可能プログラムファイル M_3 が読み込まれると、そのゲーム用実行可能プログラムファイル M_3 に定義された処理を実行するための機能構成が、メインCPU 800によってプロセスとして実現される。すなわち、図10に示すように、エンタテインメント装置700は、(1)実行可能プログラムファイルの所定領域(第一プログラム格納変数)の

格納データ(L i bファイルの正当使用によるものであれば、正当な第一プロテクトコード C_1)が格納された第一プロテクトコード格納部1000、(2)実行可能プログラムファイルの末尾領域の格納データ(L i bファイルの正当使用によるものであれば、第二プロテクトコード C_2)が格納された第二プロテクトコード格納部1001、(3)起動時にプロテクトコードチェック手続きを実行するプロテクトコードチェック処理部1002、(4)プロテクトコードチェック処理部1002からの指示にしたがって、ゲームを実行するゲーム実行処理部1003、(5)コントローラからの入力信号をゲーム実行処理部1003に渡す入力データ受付処理部1004、(6)ゲーム実行処理部1003の指示にしたがって、効果音等のオーディオ信号を生成する音声制御処理部1005、(7)ゲーム処理部1003の指示にしたがって、ゲーム画面映像のビデオ信号を生成する映像制御処理部1006、が実現される。なお、プロテクトコードチェック処理部1002には、(8)第一プロテクトコード C_1 および第二プロテクトコード C_2 を復号するコード復号処理部1002a、(9)復号後の2つのプロテクトコード C_1, C_2 の整合性をチェックするコードマッチング処理部1002b、(10)与えられた指示にしたがって、処理を終了させる終了処理部1002d、(11)コードマッチング処理部1002bのチェック結果に応じて、終了処理部1002dに処理終了指示またはゲーム処理部にゲーム実行開始指示を与える制御処理部1002c、が含まれている。

【0061】

つぎに、図11により、これらの機能構成部によって実行される処理について説明する。

【0062】

光ディスクから読み込まれた実行可能プログラムファイル M_3 が起動されると(S1100)、図10の機能構成部が以下の処理を開始する。

【0063】

ゲームの開始前に、実行可能プログラムファイル M_3 に組み込まれたL i bファイル L_2 が正当使用によるものか否かを判断するため、以下の自己診断処理がプロテクトコードチェック処理部1002によって実行される。

【 0 0 6 4 】

まず、制御処理部 1 0 0 2 c が、コード復号の指示をコード復号処理部 1 0 0 2 a に指示する。この指示に応じて、コード復号処理部 1 0 0 2 a は、第一プロテクトコード格納部 1 0 0 0 の格納データと、第二プログラムコード格納部 1 0 0 1 の格納データとを取り出し、それらをそれぞれ復号する。

【 0 0 6 5 】

フロントエンドツールの正当使用により作成された実行可能プログラムファイル M_3 であれば、1組のプロテクトコード作成に用いる関係(ここでは $C_1 = F(C_2)$)を満たすデータ C_1, C_2 が、このときの復号処理によって得られるはずである。そこで、コードマッチング処理部 1 0 0 2 b は、ここで得られた2つのデータがその関係を満たすか否かを診断し、その診断結果を制御処理部 1 1 0 2 c に通知する(S 1 1 0 1)。

【 0 0 6 6 】

そして、制御処理部 1 1 0 2 c は、実行可能プログラムファイル M_3 に組み込まれた $L i b$ ファイル L_2 が正当使用によるものか否かを、コードマッチング処理部 1 0 0 2 b の診断結果に基づき判断する(S 1 1 0 2)。具体的には、1組のプロテクトコード作成に用いた関係を満たさないという診断結果が通知された場合には、プロテクトコードに矛盾ありと判断し、1組のプロテクトコード作成に用いた関係を満たすという診断結果が通知された場合には、プロテクトコードに矛盾なしと判断する。

【 0 0 6 7 】

以上の自己診断処理によってプロテクトコードに矛盾ありと判断した場合には、制御処理部 1 1 0 2 c は、その旨を表すエラーメッセージを出力し(S 1 0 0 3)、終了処理部 1 0 0 2 d に終了指示を与える。この指示に応じて、終了処理部 1 0 0 2 d は、所定の終了処理を実行し、プログラムを終了させる(S 1 1 0 6)。

【 0 0 6 8 】

一方、プロテクトコードに矛盾なしと判断した場合には、制御処理部 1 0 0 2 d は、ゲーム実行処理部 1 0 0 3 にゲーム実行指示を与える。この指示に応じて

、ゲーム実行処理部 1 0 0 3 は、ゲーム処理を開始する (S 1 1 0 4)。

【 0 0 6 9 】

以後、コントローラを介して送られてくるプレイヤーからの指示は、入力データ受付処理部 1 0 0 4 からゲーム実行処理部 1 0 0 3 に渡される。そして、ゲーム実行処理部 1 0 0 3 は、入力データ受付処理部 1 0 0 4 から渡された指示に応じて、表示装置上の表示映像およびオーディオ装置からの音声を変化させるべく、音声制御処理部 1 0 0 5 と映像制御処理部 1 0 0 6 とを制御する。この間、ゲーム実行処理部 1 0 0 3 は、ゲームオーバーの条件を満たしたか否かの判断 (S 1 1 0 5) を定期的に行ない、ゲームオーバーと判断したら、ゲームを終了させる (S 1 1 0 6)。

【 0 0 7 0 】

このように、本実施の形態に係るエンタテインメント装置によれば、読み込んだゲーム用実行可能プログラムファイルに正当なプロテクトコードが埋め込まれている場合にだけゲームを開始させ、読み込んだゲーム用実行可能プログラムファイルに正当なプロテクトコードが埋め込まれていなければ、ゲームを終了させる。したがって、フロントエンドツールの正当使用により、ゲーム用実行可能プログラムファイルが作成されていなければ、エンタテインメント装置でゲームをすることができなくなる。このため、フロントエンドツールの正当使用者以外のユーザによる L i b ファイル使用を止すことできる。

【 0 0 7 1 】

なお、以上においてはエンタテインメント装置を例に挙げたが、上述の実行可能プログラムファイル作成処理により作成された実行可能プログラムファイルを実行するための装置は、必ずしもエンタテインメント装置である必要はない。例えば、プログラムの実行能力のある、一般的な電子計算機であってもよい。もちろん、実行可能プログラムファイルも、ゲームを実行させるものである必要はない。

【 0 0 7 2 】

また、ここでは、記憶媒体から実行可能プログラムファイルが実行装置に読み込まれるようにしているが、ネットワーク等の伝送媒体によって実行可能プログ

ラムファイルが伝送され、それが実行装置のハードディスクに格納されるようにしてもよい。

【 0 0 7 3 】

また、以上においては、ライブラリファイルをプロテクト対象としているが、その他のファイル(例えば、画像データ等が格納されたファイル等の、実行可能ファイル以外のファイル)をプロテクト対象としてもよい

【 0 0 7 4 】

【発明の効果】

本発明によれば、ライブラリ等、実行可能ファイル以外のファイルの不正コピーを防止することができる。

【図面の簡単な説明】

【図 1】

本発明の実施の一形態に係る情報処理システムのハードウェア構成図である。

【図 2】

本発明の実施の一形態に係る情報処理システムの機能構成図である。

【図 3】

本発明の実施の一形態に係る実行可能プログラム作成処理のフローチャートである。

【図 4】

本発明の実施の一形態に係る 1 組のプロテクトコードの生成方法を概念的に示した図である。

【図 5】

本発明の実施の一形態に係る実施可能プログラムファイルが作成されるまでの処理を概念的に示した図である。

【図 6】

本発明の実施の一形態に係る実施可能プログラムファイルのデータ構造を概念的に示した図である。

【図 7】

本発明の実施の一形態に係るエンタテインメント装置およびその周辺機器の外

観図である。

【図 8】

本発明の実施の一形態に係るエンタテインメント装置のハードウェア構成図である。

【図 9】

本発明の実施の一形態に係る光ディスクの格納データのデータ構造を概念的に示した図である。

【図 1 0】

本発明の実施の一形態に係るエンタテインメント装置の機能構成図である。

【図 1 1】

本発明の実施の一形態に係る実行可能プログラム作成処理のフローチャートである。

【符号の説明】

L ₁ …暗号化ライブラリファイル	L ₂ …ライブラリファイル
M ₁ …ソースファイル	M ₂ …オブジェクトファイル
M ₃ …実行可能ファイル	
1 0 0 …情報処理装置	1 0 1 …C P U
1 0 2 …グラフィックス機構	1 0 3 …主記憶
1 0 4 …メモリバス制御チップ	1 0 5 , 1 0 6 , 1 0 9 …バス
1 0 7 …ハードディスクドライブ	1 0 8 …ハードディスク
1 1 0 …ブリッジ回路	1 1 1 …入力装置コントローラ
1 1 2 …C D - R O M ドライブ	1 1 3 …出力装置コントローラ
1 2 0 …入力装置	1 3 0 …出力装置
2 0 0 …入力受付部	2 0 1 …起動時チェック処理部
2 0 2 …開発成果記憶部	2 0 3 …コンパイル処理部
2 0 4 …ライブラリ記憶部	2 0 5 …復号化処理部
2 0 6 …一時ファイル記憶部	2 0 7 …リンク処理部
2 0 8 …一時ファイル削除処理部	2 0 9 …コード付与処理部

2 1 0 …制御処理部

7 0 0 …エンタテインメント装置

8 0 0 …メインCPU

8 0 1 …グラフィックプロセッサ

8 0 2 …I/Oプロセッサ

8 0 3 …光ディスク制御部

8 0 4 …サウンド再生処理プロセッサ

8 0 5 …サウンドバッファ

8 0 6 …OS-ROM

8 0 7 …IOPメモリ

8 0 8 …メインメモリ

8 0 9 …バス

1 0 0 0 …第一プロテクトコード格納部

1 0 0 1 …第二プロテクトコード格納部

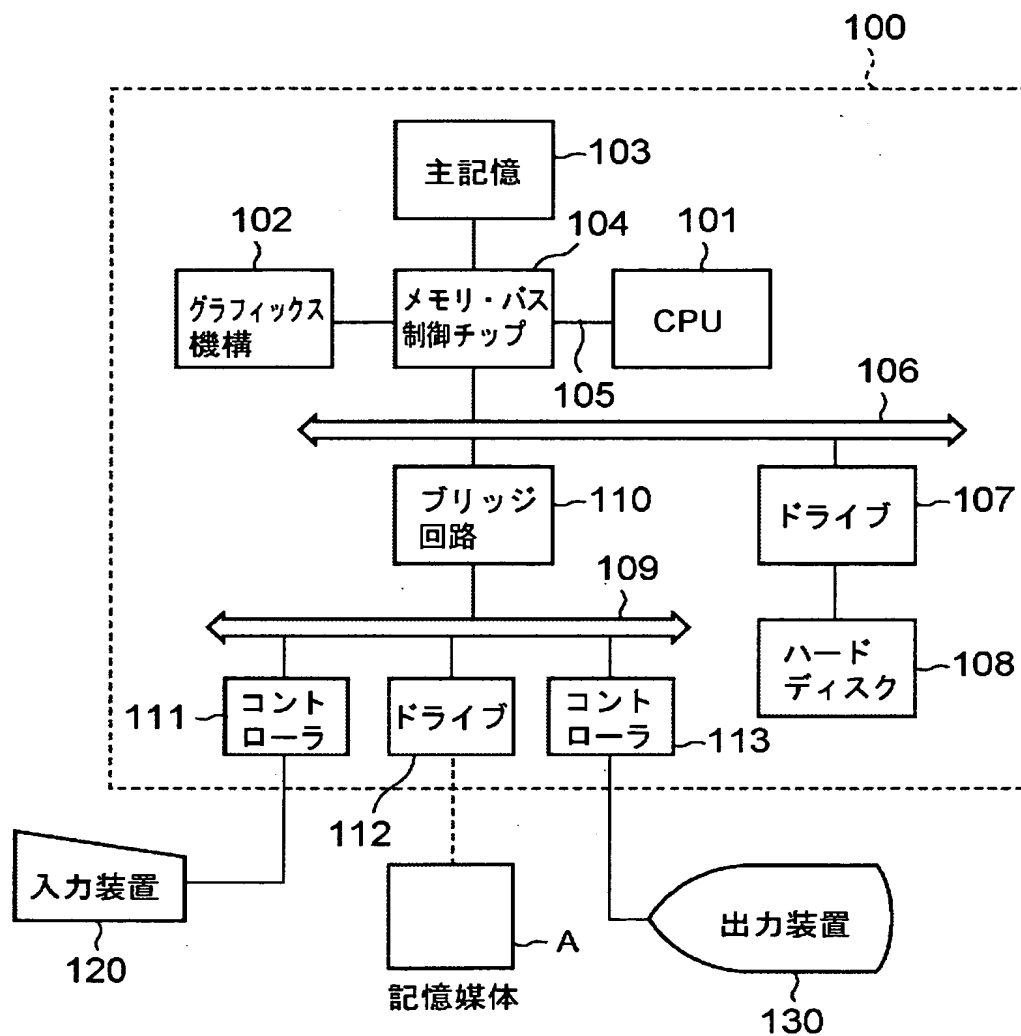
1 0 0 2 …プロテクトコードチェック処理部

1 0 0 3 …ゲーム実行処理部

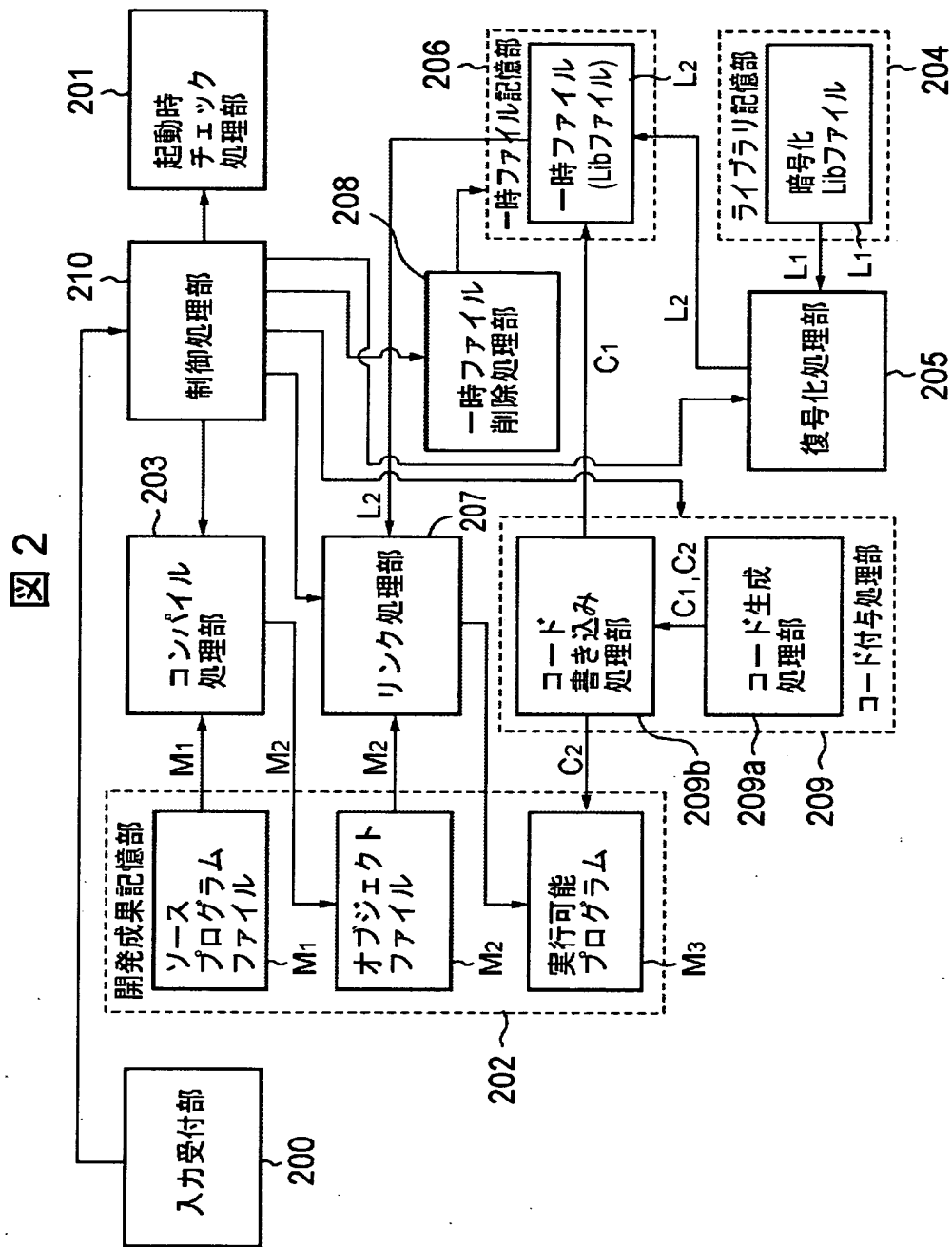
【書類名】 図面

【図 1】

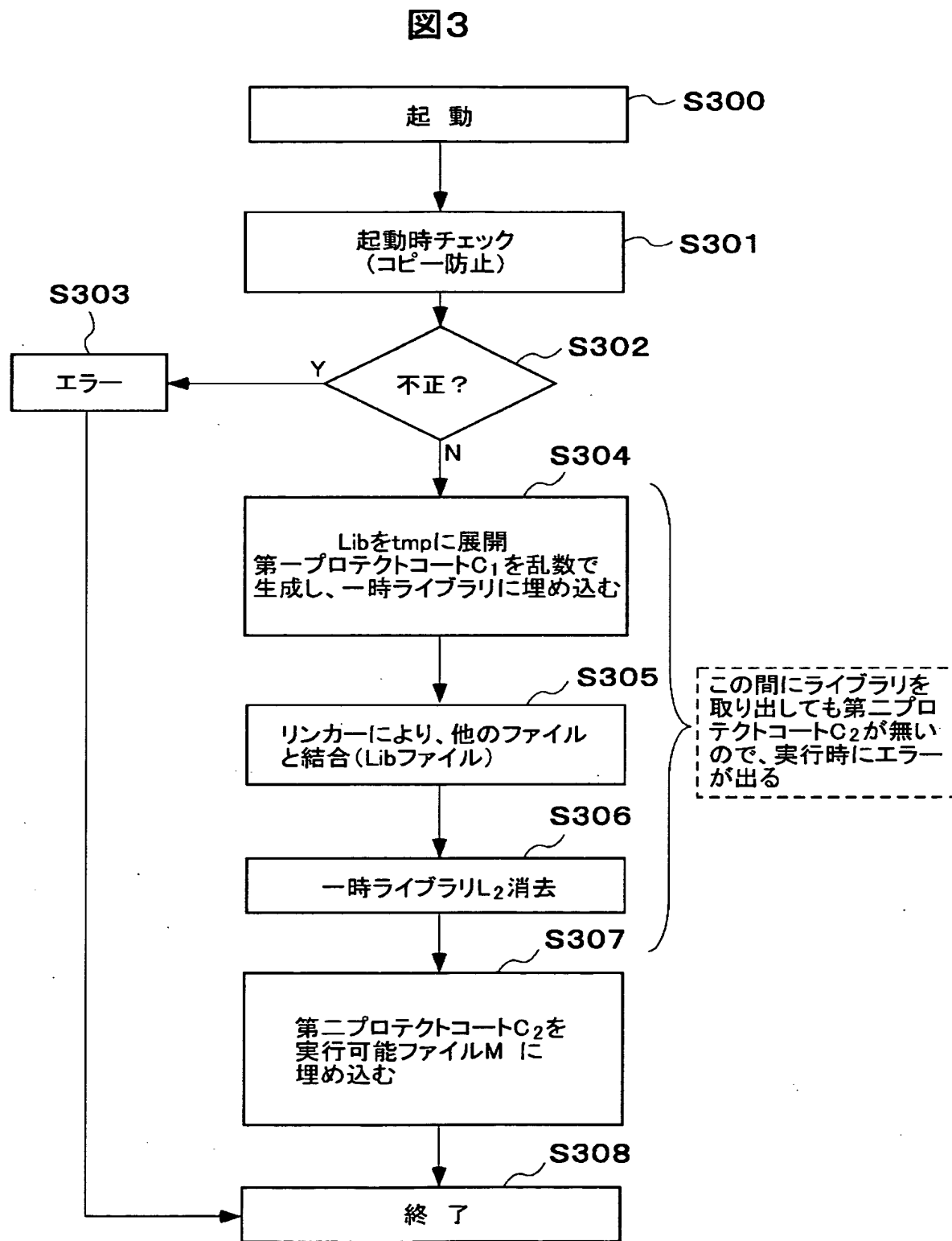
図 1



【図 2】

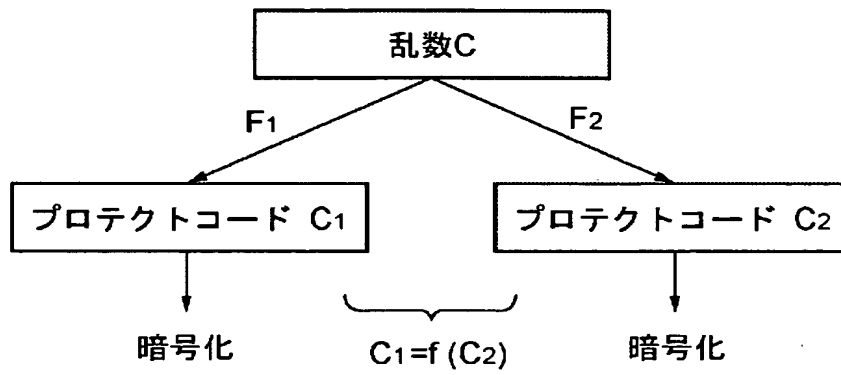


【図 3】



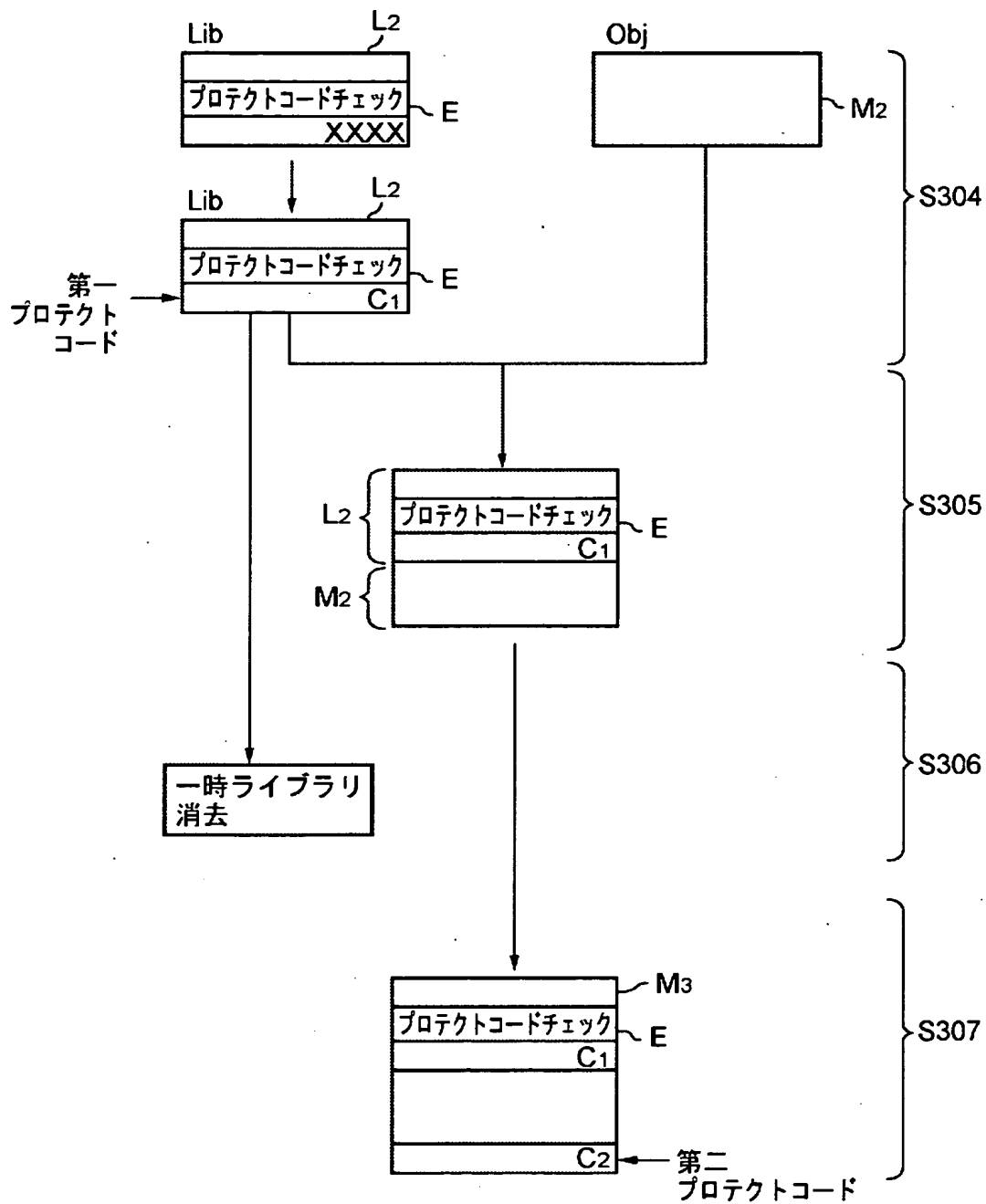
【図 4】

図 4



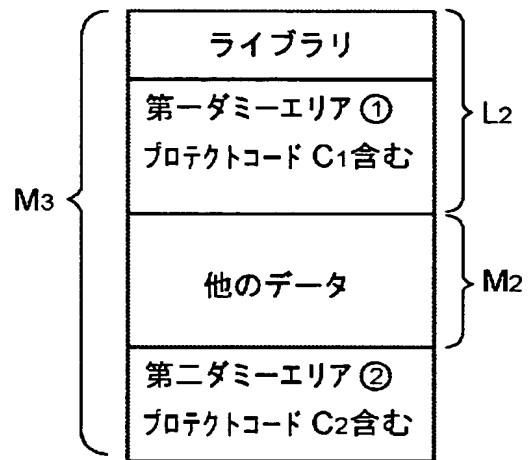
【図 5】

図 5



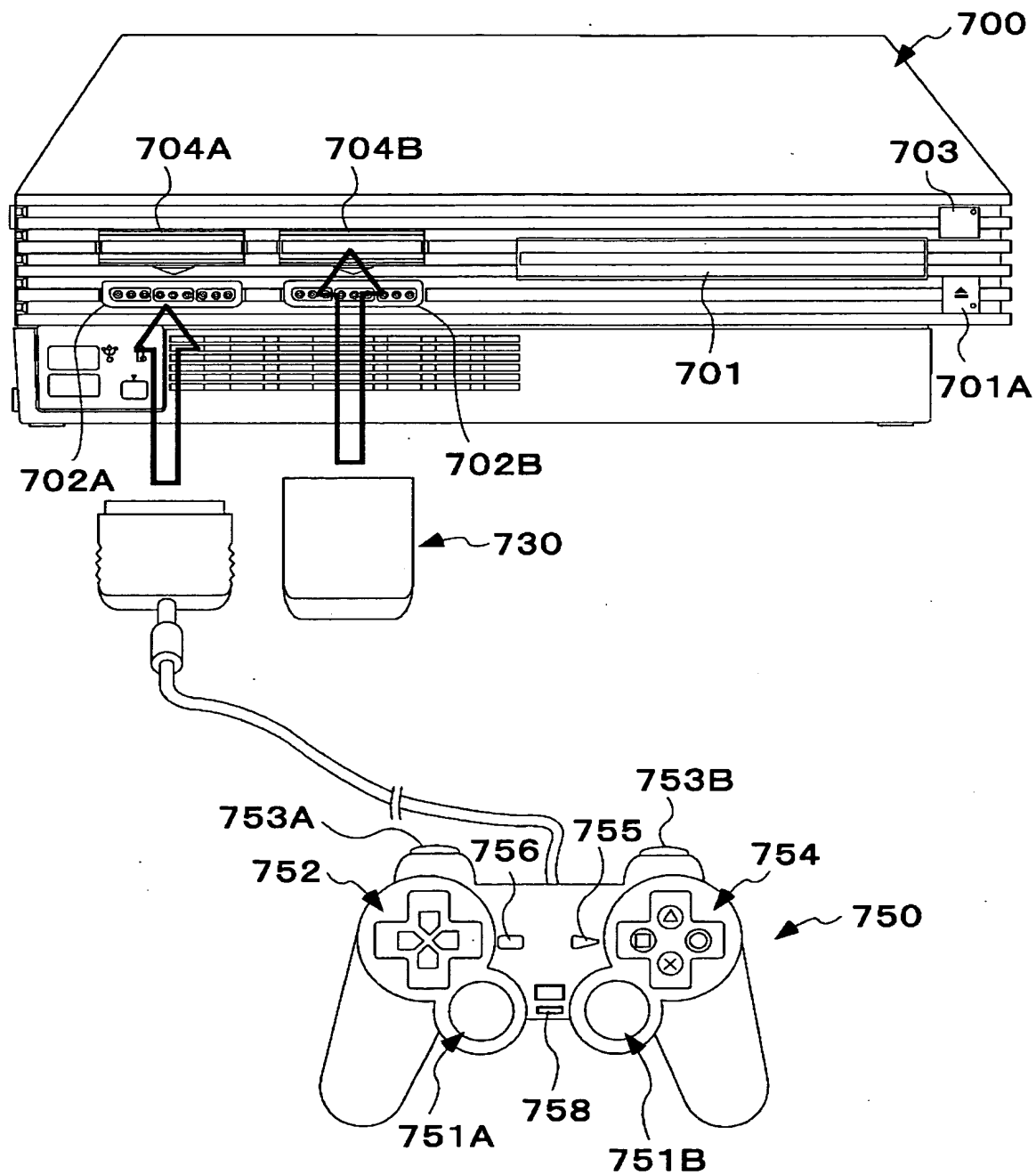
【図 6】

図 6

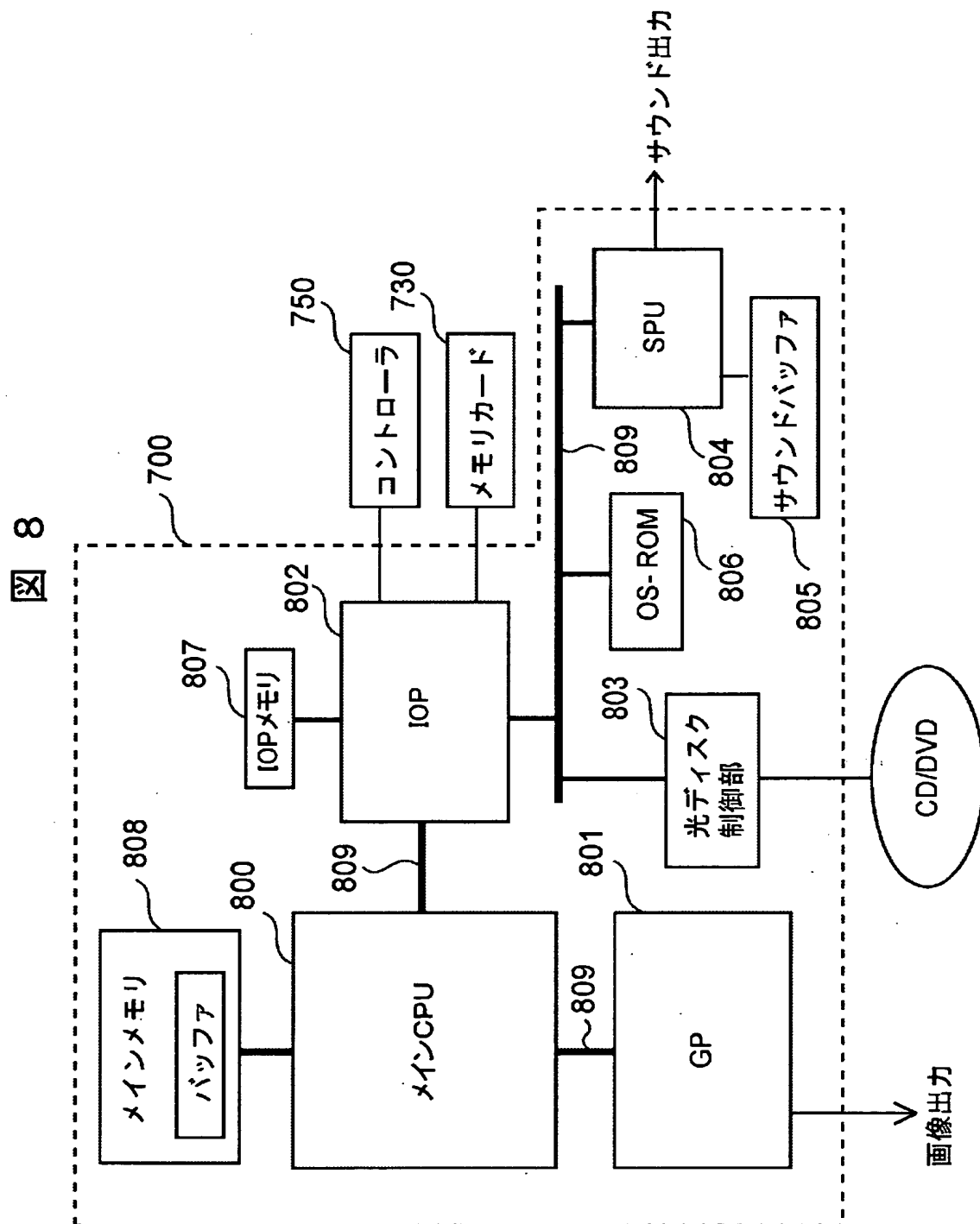


【図 7】

図 7

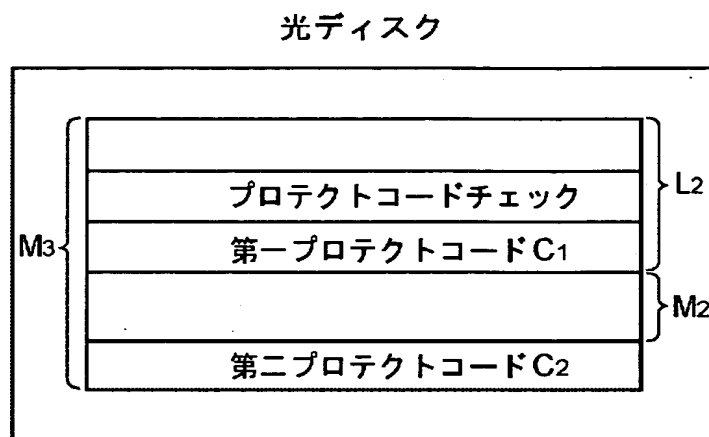


【図 8】



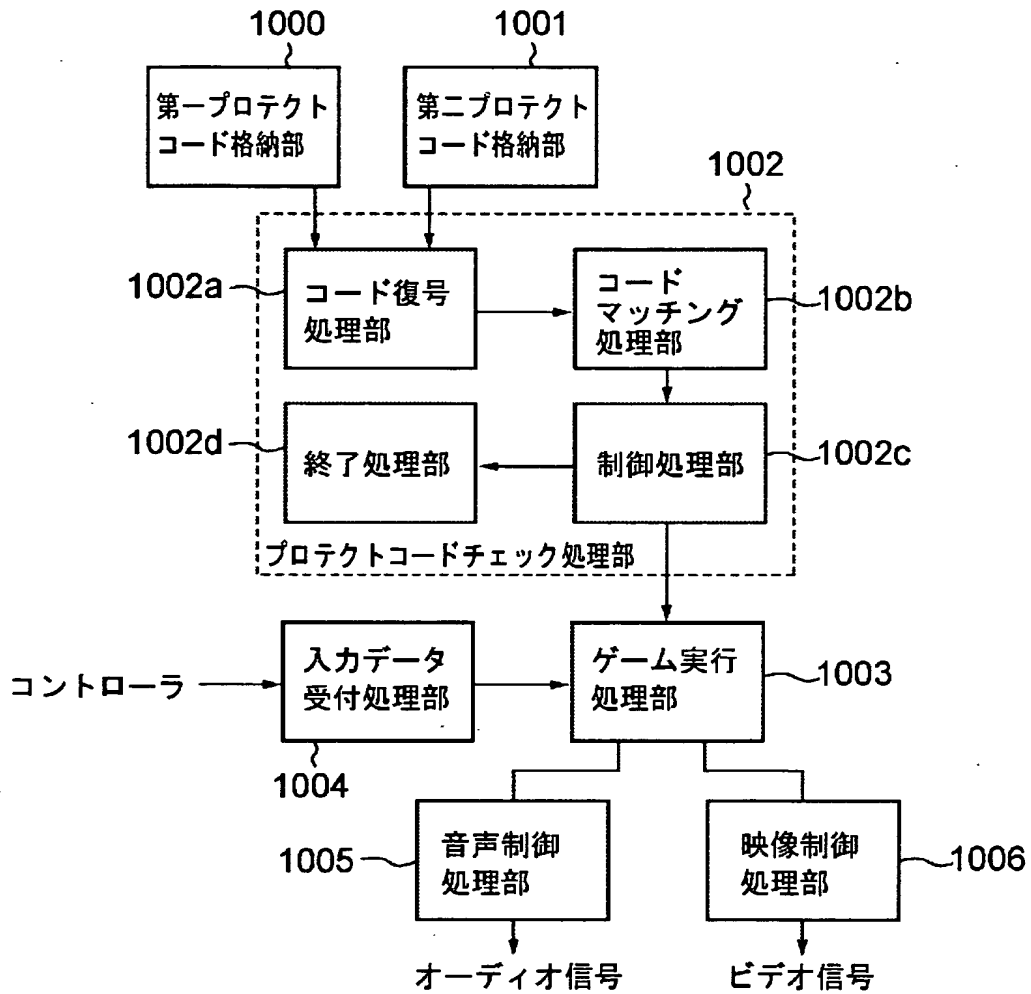
【図 9】

図 9



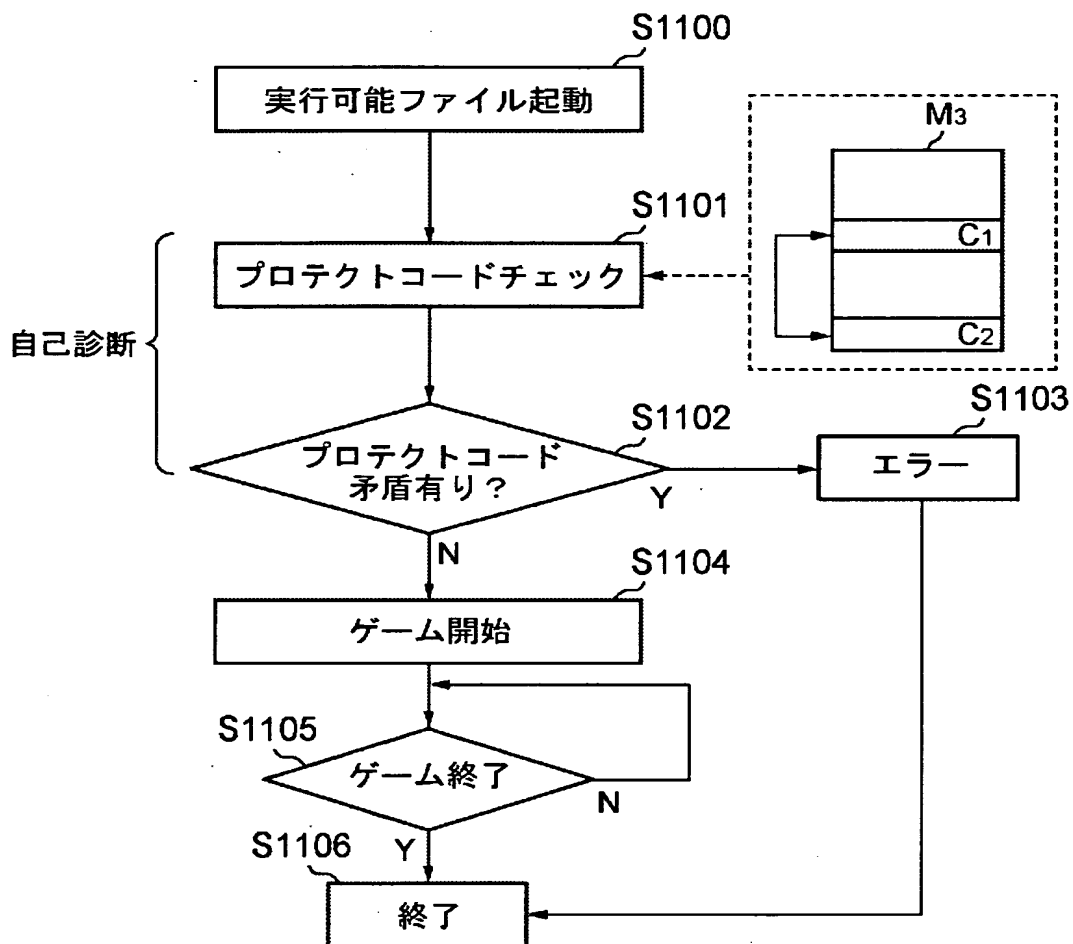
【図 1 0】

図 10



【図 11】

図 11



【書類名】 要約書

【課題】 ライブラリ等の不正コピーを防止する。

【解決手段】 暗号化ライブラリファイル L_2 には、実行可能プログラムファイルに埋め込まれた2つのプロテクトコードに矛盾があれば、初期化段階で処理を終了させるプロテクトコードチェック手続きが含まれている。実行可能プログラムファイル作成時、暗号化ライブラリファイル L_2 が復号され、それに第一プロテクトコード C_1 が埋め込まれる(S 3 0 4)。そのあと、復号後のライブラリファイル L_2 とオブジェクトファイル M_2 とが結合編集され、実行可能プログラムファイルが作成される(S 3 0 5)。そして、使用済みライブラリファイル L_2 を消去してから(S 3 0 6)、実行可能プログラムファイル M_3 に、第一プロテクトコード C_1 と所定の関係にある第二プロテクトコード C_2 がさらに埋め込まれる(S 3 0 7)。

【選択図】 図 5

出 願 人 履 歴 情 報

識別番号 [395015319]

1. 変更年月日 1997年 3月31日
[変更理由] 住所変更
住 所 東京都港区赤坂7-1-1
氏 名 株式会社ソニー・コンピュータエンタテインメント